



A conservative coupling algorithm between a compressible flow and a rigid body using an Embedded Boundary method

Laurent Monasse, Virginie Daru, Christian Mariotti, Serge Piperno, Christian Tenaud

► To cite this version:

Laurent Monasse, Virginie Daru, Christian Mariotti, Serge Piperno, Christian Tenaud. A conservative coupling algorithm between a compressible flow and a rigid body using an Embedded Boundary method. *Journal of Computational Physics*, 2012, 231, pp.2977-2994. 10.1016/j.jcp.2012.01.002 . hal-01163118

HAL Id: hal-01163118

<https://hal.science/hal-01163118>

Submitted on 12 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A conservative coupling algorithm between a compressible flow and a rigid body using an Embedded Boundary method

Laurent Monasse^{1, 2, 3}, Virginie Daru^{3, 4}, Christian Mariotti¹, Serge Piperno²,
Christian Tenaud³

¹ CEA-DAM-DIF 91297 Arpajon, France
email: {laurent.monasse, christian.mariotti}@cea.fr

² Université Paris-Est, CERMICS (ENPC),
77455 Marne la Vallée cedex, France
email: {monassel, piperno}@cermics.enpc.fr

³ LIMSI-CNRS, 91403 Orsay, France
email: {laurent.monasse, virginie.daru, christian.tenaud}@limsi.fr

⁴ Lab. DynFluid, Ensam, 75013 Paris, France
email: virginie.daru@ensam.eu

ABSTRACT

This paper deals with a new solid-fluid coupling algorithm between a rigid body and an unsteady compressible fluid flow, using an Embedded Boundary method. The coupling with a rigid body is a first step towards the coupling with a Discrete Element method. The flow is computed using a Finite Volume approach on a Cartesian grid. The expression of numerical fluxes does not affect the general coupling algorithm and we use a one-step high-order scheme proposed by Daru and Tenaud [Daru V, Tenaud C., J. Comput. Phys. 2004]. The Embedded Boundary method is used to integrate the presence of a solid boundary in the fluid. The coupling algorithm is totally explicit and ensures exact mass conservation and a balance of momentum and energy between the fluid and the solid. It is shown that the scheme preserves uniform movement of both fluid and solid and introduces no numerical boundary roughness. The efficiency of the method is demonstrated on challenging one- and two-dimensional benchmarks.

Key Words: Compressible flows; Shock capturing scheme; Discrete Element method; Fluid-structure interaction; Embedded Boundary method

1 Introduction

This work is devoted to the development of a coupling method for fluid-structure interaction in the compressible case. We intend to simulate transient dynamics problems, such as the impact of shock waves onto a structure, with possible fracturing causing the ultimate breaking of the structure. An inviscid fluid flow model is considered, being convenient for treating such short time scale phenomena. The simulation of fluid-structure interaction problems is often computationally challenging due to the generally different numerical methods used for solids and fluids and the instability that may occur when coupling these methods. Monolithic methods have been employed, using an Eulerian formulation for both the solid and the fluid (for instance, the diffusive interface method [16, 1]), or a Lagrangian formulation for both the fluid and the solid (for example, the PFEM method [26]), but in general, most solid solvers use Lagrangian formulations and fluid solvers use Eulerian formulations. In this paper we consider the coupling of a Lagrangian solid solver with an Eulerian fluid solver.

For the coupling in space, a possible choice is to deform the fluid domain in order to follow the movement of the solid boundary: the Arbitrary Lagrangian-Eulerian (ALE) method has been developed to that end. It has been widely used for incompressible [11, 19] and compressible [15] fluid-structure interaction. However, when solid impact or fracture occur, ALE methods are faced with a change of topology in the fluid domain that requires remeshing and projection of the fluid state on the new mesh, which are costly and error prone procedures. Moreover remeshing is poorly adapted to load balancing for parallel computations.

In order to allow for easier fracturing of the solid, we instead choose a method based on fictitious domains that solves the fluid flow on a fixed Eulerian mesh, on which a Lagrangian solid body is superimposed. A special treatment is then applied on fluid cells near the boundary and inside the solid. Different types of fictitious domain methods have been developed over the last thirty years. They can roughly be classified in three main classes: penalization methods, interpolation methods and conservative methods. Among penalization methods, the Immersed Boundary method is certainly the best known and most widely used for fluid-structure interaction. It was originally introduced by Peskin for incompressible blood flows [44, 45]. The solid boundaries deform under the action of the fluid velocity, and the presence of the solid adds forces in the fluid formulation that enforce the impermeability of the solid. However, Xu and Wang have pointed out some numerical leaking of fluid into the solid [49]. Following Leveque and Li [33, 34], they advocate the use of the Immersed Interface method, which incorporates jump conditions in the finite differences used. However, the absence of fluid mass loss is still not ensured exactly. In a different approach, Olovsson *et al.* [40, 2] couple an Eulerian and a Lagrangian method by penalizing the penetration of the solid into the fluid by a damped spring force. As the stiffness of the spring goes to infinity, the penetration goes to zero. Boiron *et al.* [5] and Paccou *et al.* [41] consider the solid as a porous medium, using a Brinkman porosity model. As the porosity goes to zero, the solid becomes impermeable. However, in both cases, as the stiffness grows or the porosity decreases, the use of implicit schemes is mandatory to avoid the severe stability condition of explicit schemes [5, 41]. For the high speed phenomena we consider, we use explicit solid and fluid solvers and an explicit coupling algorithm is better suited in order to avoid costly iterative procedures.

A second class of fictitious domain methods consists in enforcing the boundary conditions through interpolations in the vicinity of the boundary, using the exact values taken by the fluid on the boundary [37, 13]. The method seems to be very versatile, being used with incompressible Navier-Stokes [37, 13], Reynolds-averaged Navier-Stokes [10, 29, 30], turbulent boundary layer laws [6] and compressible Navier-Stokes [42]. The Ghost Fluid method developed by Fedkiw *et al.* [18, 17] relies on the same type of principle for compressible fluids. The interface is tracked using a level-set function, and conditions are applied on both sides of the interface to interpolate the boundary conditions. The advantage of these methods is that they do not suffer from additional time-step restriction due to stability, and the order of accuracy of the boundary conditions can be set *a priori*. However, the interpolation does not ensure the conservation of mass, momentum and energy in the system. This can cause problems when dealing with shock waves interacting with solids.

In this article, we rather consider the third class of conservative fictitious domain methods, which seems the most adequate framework to develop our coupling algorithm. These methods are generally referred to as Embedded Boundary methods, and they rely on a modified integration of the numerical fluid fluxes in the cells cut by the solid boundary [43, 14, 22, 25]. The original idea of the method can be traced back to Noh’s CEL code [39]. The new contribution of the present work consists in the coupling algorithm and its properties. The Embedded Boundary method that we use is essentially identical to previous works [43, 14, 22, 25]. The different versions of the Embedded Boundary method mainly differ in the way the stability condition arising from small cut-cells is enforced and we develop here a slightly different procedure in order to deal with solid boundaries coming close to each other. The method can be implemented independently from the time integration scheme used for the fluid, whether based on space-time splitting or multi-level time integration. Conservative fictitious domain methods have proven to give satisfactory conservation results for inviscid compressible flows in the case of static solid boundaries. Nevertheless, to our knowledge, conservation issues of the coupling have not been studied in the case of moving solids. We establish new conservation results in such a case. Our coupling method is designed to be capable of treating the general case of moving deformable bodies. In the present work, however, we only consider non-deformable (rigid) solid bodies. The case of deformable bodies is the object of ongoing work.

The fluid and solid solvers that we consider were chosen according to their ability to deal with shock waves and fracturing solids. The solid solver is based on a Discrete Element method, implemented in a code named Mka3D in the CEA [35]. It can handle elasticity as well as fracture and impact of solids. Solids are discretized into polyhedral particles, which interact through well-designed forces and torques. The particles have a rigid-body motion, and fracture is treated in a straightforward way by removing the physical cohesion between particles. The work reported in this article is a first step towards the coupling with the Mka3D code. The time integration scheme used by Mka3D (Verlet for displacement of the center of mass and RATTLE for rotation [38]) is retained for the rigid body treatment. Concerning the fluid solver, we use a Cartesian grid explicit finite volume method, based on the high-order one-step

monotonicity-preserving scheme developed in [8] and space–time splitting. However we emphasize that our coupling method is independent from both the Discrete Element method (as long as a solid interface is defined) and the numerical scheme used for the fluid calculation.

The article is organized as follows: we first present briefly the solid and fluid methods in section 2. In sections 3 and 4, we describe the proposed explicit coupling procedure between the fluid and the moving solid in the framework of an Embedded Boundary method. The analysis of the conservation properties of the coupling is reported in section 5, where we show that mass, momentum and energy of the solid-fluid system are exactly preserved. In section 6, we demonstrate results about the preservation on a discrete level of two solid-fluid systems in uniform movement. Finally, we illustrate the efficiency and accuracy of the method on one and two-dimensional static and dynamic benchmarks in section 7.

2 Solid and fluid discretization methods

2.1 Solid time-discretization method

We consider a non-deformable solid (rigid body). The position and velocity of the solid are given, respectively, by the position of its center of mass \mathbf{X} , the rotation matrix \mathbf{Q} , the velocity of the center of mass \mathbf{V} and the angular momentum matrix \mathbf{P} . The physical characteristics of the solid are its mass m and its matrix of inertia \mathbf{R} which, in the inertial frame, is a diagonal matrix with the principal moments of inertia I_1 , I_2 and I_3 on the diagonal. Here, we instead use the diagonal matrix $\mathbf{D} = \text{diag}(d_1, d_2, d_3)$, where:

$$\forall i \in \{1, 2, 3\}, d_i = \frac{I_1 + I_2 + I_3}{2} - I_i.$$

The angular momentum matrix \mathbf{P} can be related to the usual angular velocity vector $\boldsymbol{\Omega}$ by the relation $\mathbf{P} = \mathbf{D}\mathbf{j}(\boldsymbol{\Omega})\mathbf{Q}$, where the map $\mathbf{j} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ is defined such that:

$$\forall \mathbf{x} \in \mathbb{R}^3, \forall \mathbf{y} \in \mathbb{R}^3, \mathbf{j}(\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \wedge \mathbf{y}.$$

Let us denote by \mathbf{F} and \mathcal{M} the external forces and torques acting on the solid, and by Δt the time-step. In order to preserve the energy of the solid over time integration of the movement, we choose a symplectic second-order scheme for constrained Hamiltonian systems, the RATTLE scheme [23]:

$$\mathbf{V}^{n+\frac{1}{2}} = \mathbf{V}^n + \frac{\Delta t}{2m} \mathbf{F}^n, \quad (1)$$

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \mathbf{V}^{n+\frac{1}{2}}, \quad (2)$$

$$\mathbf{P}^{n+\frac{1}{2}} = \mathbf{P}^n + \frac{\Delta t}{4} \mathbf{j}(\mathcal{M}^n) \mathbf{Q}^n + \frac{\Delta t}{2} \Lambda^n \mathbf{Q}^n, \quad (3)$$

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta t \mathbf{P}^{n+\frac{1}{2}} \mathbf{D}^{-1}, \quad (4)$$

$$\text{with } \Lambda^n \text{ such that } (\mathbf{Q}^{n+1})^T \mathbf{Q}^{n+1} = \mathbf{I}, \quad (5)$$

$$\mathbf{V}^{n+1} = \mathbf{V}^{n+\frac{1}{2}} + \frac{\Delta t}{2m} \mathbf{F}^{n+1}, \quad (6)$$

$$\mathbf{P}^{n+1} = \mathbf{P}^{n+\frac{1}{2}} + \frac{\Delta t}{4} \mathbf{j}(\mathcal{M}^{n+1}) \mathbf{Q}^{n+1} + \frac{\Delta t}{2} \tilde{\Lambda}^{n+1} \mathbf{Q}^{n+1}, \quad (7)$$

$$\text{with } \tilde{\Lambda}^{n+1} \text{ such that } (\mathbf{Q}^{n+1})^T \mathbf{P}^{n+1} \mathbf{D}^{-1} + \mathbf{D}^{-1} (\mathbf{P}^{n+1})^T \mathbf{Q}^{n+1} = \mathbf{0}. \quad (8)$$

The symmetric matrices Λ^n and $\tilde{\Lambda}^{n+1}$ play the role of Lagrange multipliers for the constraints on matrices \mathbf{Q}^{n+1} and \mathbf{P}^{n+1} .

The scheme makes use of the velocity at half time-step $\mathbf{V}^{n+\frac{1}{2}}$, which is constant during the time-step. Let us now consider the angular velocity. For a rigid solid, we have for all points \mathbf{x} :

$$\mathbf{X} - \mathbf{x} = \mathbf{Q} \cdot (\mathbf{X}^0 - \mathbf{x}^0),$$

\mathbf{X}^0 and \mathbf{x}^0 being material points of the solid at initial time. Using the identity $\boldsymbol{\Omega} \wedge (\mathbf{Q}\mathbf{x}) = \mathbf{P}\mathbf{D}^{-1}\mathbf{x}$ for all \mathbf{x} , the velocity at point \mathbf{x} can be written as:

$$\mathbf{V}(\mathbf{x}) = \mathbf{V} + \mathbf{P}\mathbf{D}^{-1} \cdot (\mathbf{X}^0 - \mathbf{x}^0)$$

which is more convenient for use in the time scheme. In analogy with displacement, we consider $\mathbf{P}^{n+\frac{1}{2}}$ as constant during the time-step, and we define the velocity of point \mathbf{x} at half time-step $(n + \frac{1}{2})\Delta t$:

$$\mathbf{V}^{n+\frac{1}{2}}(\mathbf{x}) = \mathbf{V}^{n+\frac{1}{2}} + \mathbf{P}^{n+\frac{1}{2}} \mathbf{D}^{-1} \cdot (\mathbf{X}^0 - \mathbf{x}^0).$$

2.2 Fluid discretization method

The problem of the interaction of shock waves with solid surfaces can be at first studied using an inviscid fluid model. In this work, we consider inviscid compressible flows, which follow the Euler equations:

$$\mathbf{w}_t + \nabla \cdot \mathbf{f}(\mathbf{w}) = \mathbf{0},$$

where $\mathbf{w} = (\rho, \rho \mathbf{u}, \rho E)^T$ is the vector of the conservative variables, and $\mathbf{f}(\mathbf{w})$ is the Euler flux:

$$\mathbf{f} = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I} \\ (\rho E + p) \mathbf{u} \end{pmatrix},$$

where the pressure p is given by a perfect gas law: $p = (\gamma - 1) (\rho E - \frac{1}{2} \rho \mathbf{u} \cdot \mathbf{u})$.

To solve these equations, we use the OSMP numerical scheme, which is a one-step high-order scheme developed in [8, 9]. It is derived using a coupled space-time Lax-Wendroff approach, where the formal order of accuracy in the scalar case can be set at arbitrary order (in this paper, we use order 11, that is the OSMP11 scheme). Imposing the MP conditions (Monotonicity Preserving) prevents the appearance of numerical oscillations in the vicinity of discontinuities while simultaneously avoiding the numerical diffusion of extrema. In one space dimension, on a uniform mesh with step-size Δx , at order p , it can be written:

$$w_j^{n+1} = w_j^n - \frac{\Delta t}{\Delta x} (f_{j+\frac{1}{2}}^p - f_{j-\frac{1}{2}}^p)$$

where $f_{j+\frac{1}{2}}^p$ is the p th-order-accurate numerical flux of the scheme at the cell interface $(j + \frac{1}{2})$. Given the l eigenvectors of the Jacobian matrix of the flux \mathbf{r}_k and eigenvalues λ_k , the general expression of the numerical fluxes can be written :

$$f_{j+\frac{1}{2}}^p = f_{j+\frac{1}{2}}^{\text{Roe}} + \frac{1}{2} \sum_k (\psi^p \mathbf{r})_{k,j+\frac{1}{2}} \quad (9)$$

where, for clarity, the superscript n has been omitted. $f_{j+\frac{1}{2}}^{\text{Roe}}$ is the first order Roe flux defined as follows:

$$f_{j+\frac{1}{2}}^{\text{Roe}} = \frac{1}{2} (f_j + f_{j+1}) - \frac{1}{2} \sum_k (\delta |f| \mathbf{r})_{k,j+\frac{1}{2}} \quad (10)$$

with

$$\delta |f|_{k,j+1/2} = |\lambda|_{k,j+1/2} \delta \alpha_{k,j+1/2}$$

$\delta \alpha_{k,j+\frac{1}{2}} = \mathbf{r}_k \cdot (w_{j+1}^n - w_j^n)$ being the k -th Riemann invariant of the Jacobian matrix. The ψ^p are corrective terms to obtain order p . The function ψ can be decomposed in odd and even parts:

$$\psi_{k,j+\frac{1}{2}}^p = \sum_{n=1}^m \psi_{k,j+\frac{1}{2}}^{2n} + js \sum_{n=1}^{m1} \psi_{k,j+1-\frac{js}{2}}^{2n+1} \quad (11)$$

where $m = \lfloor \frac{p}{2} \rfloor$, $m1 = \lfloor \frac{(p-1)}{2} \rfloor$ ($\lfloor \cdot \rfloor$ is the integer division symbol), and $js = \text{sign}(\lambda_{k,j+\frac{1}{2}})$. The odd and even ψ functions are given by the recurrence formulae (valid for $n \geq 1$):

$$\psi_{k,j+\frac{1}{2}}^{2n} = \sum_{l=0}^{2n-2} (-1)^l \mathbf{C}_{2n-2}^l \cdot (c_{2n} \delta \alpha)_{k,j+\frac{1}{2}+n-1-l} \quad (12)$$

$$\psi_{k,j+\frac{1}{2}}^{2n+1} = \sum_{l=0}^{2n-1} (-1)^l \mathbf{C}_{2n-1}^l \cdot (c_{2n+1} \delta \alpha)_{k,j+\frac{1}{2}+(n-1-l) \cdot js}, \quad (13)$$

where $\mathbf{C}_r^s = \frac{r!}{[(r-s)!s!]}$. The coefficients c_q depend on the local CFL number, $\nu_{k,j+\frac{1}{2}} = \frac{\delta t}{\delta x} \lambda_{k,j+\frac{1}{2}}$, and are given by:

$$\begin{aligned} (c_2)_{k,j+\frac{1}{2}} &= |\lambda|_{k,j+\frac{1}{2}} (1 - |\nu|_{k,j+\frac{1}{2}}) \\ (c_{q+1})_{k,j+\frac{1}{2}} &= \frac{|\nu|_{k,j+\frac{1}{2}} + (-1)^q \lfloor \frac{q+1}{2} \rfloor}{q+1} \cdot (c_q)_{k,j+\frac{1}{2}}, \quad q \geq 2 \end{aligned} \quad (14)$$

At order p , the stencil of the scheme uses $p+2$ points. Flux limiting TVD or MP constraints are then be applied to ψ^p to make the scheme non-oscillatory. The detail of the limiting procedure can be found in [9].

Near cut-cells, the existence of an adequate stencil of fluid points is not necessarily provided for. Two main types of solutions can be devised: either lower the order of accuracy and thus the stencil width, or construct fictitious fluid values in the solid. We resort to the second solution, with simple mirroring conditions with respect to the solid boundary. The solution is satisfactory as long as the solid is larger than the stencil of the scheme, which is the case for the numerical examples considered in this paper. In case this condition fails, we could resort to Ghost Fluid-type methods as in [18].

In two dimensions, the fluxes are computed using a directional Strang splitting [47] which is second-order accurate. However the error of the scheme remains very low [8]. This splitting procedure will be expressed in section 4 devoted to the coupling algorithm.

3 Treatment of the cells cut by the solid boundary in the Embedded Boundary method

In this section, we recall the main ideas of the Embedded Boundary method as exposed in [43, 14, 25].

In order to take into account the position of the solid in the fluid domain, we rely on the Embedded Boundary method, which consists in modifying the fluid fluxes in cells that are cut by the solid boundary (named cut cells), as in [25, 14]. At time t , for a cut cell \mathcal{C} , we assume that the solid occupies a volume fraction $\alpha_{\mathcal{C}}$. We also assume that the density, velocity and pressure are constant in the cell. The fluid mass, momentum and energy quantities contained in the cell are therefore equal to their value at the center of the cell times the volume of the cell and the volume fraction of fluid $1 - \alpha_{\mathcal{C}}$. In the same way, the computed fluxes are assumed to be constant on the faces of a cell. Denoting by $\kappa_{\mathcal{C}_1\mathcal{C}_2}$ the solid surface fraction of the face between cells \mathcal{C}_1 and \mathcal{C}_2 , the effective flux between \mathcal{C}_1 and \mathcal{C}_2 is the computed flux times the surface of their interface times the fluid surface fraction $1 - \kappa_{\mathcal{C}_1\mathcal{C}_2}$. Additional fluxes come from the presence of the moving solid boundary. These fluxes arise due to the change in surface fractions and the work of the fluid pressure on the solid surface. They are expressed in order to yield exact conservation of fluid mass and of the total momentum and energy of the system.

For the sake of simplicity, we limit ourselves to two space dimensions. However the three-dimensional case can be carried out in a similar way. Let us consider a fluid cell \mathcal{C} cut by the boundary, as shown in Figure 1. The indices l , r , t and b indicate respectively left, right, top and bottom in the sequel. Integrating the Euler equations on the cut cell and over the time interval $[n\Delta t, (n+1)\Delta t]$, and applying the divergence theorem, we get:

$$\begin{aligned} (1 - \alpha_{\mathcal{C}}^{n+1})\Delta w_{\mathcal{C}} &= \Delta t \left(\frac{1 - \kappa_{\mathcal{C}l}}{\Delta x} f_{cl} - \frac{1 - \kappa_{\mathcal{C}r}}{\Delta x} f_{cr} + \frac{1 - \kappa_{\mathcal{C}b}}{\Delta y} f_{cb} \right. \\ &\quad \left. - \frac{1 - \kappa_{\mathcal{C}t}}{\Delta y} f_{ct} \right) + \frac{\Delta t}{\Delta x \Delta y} X_{\mathcal{F}} + \sum_{\mathcal{F} \in \mathcal{C}} \Delta w_{\mathcal{F}}^n \end{aligned} \quad (15)$$

where $\Delta w_{\mathcal{C}} = w_{\mathcal{C}}^{n+1} - w_{\mathcal{C}}^n$ is the time increment and all fluxes are time-averaged over the time interval (the time averaging will be specified later). At the solid walls, pressure forces cause momentum and energy exchange between the solid and the fluid. They are taken into account through the exchange term $X_{\mathcal{F}}$. The detailed expression of $X_{\mathcal{F}}$ will be given in section 4.4. Finally, the quantity $\Delta w_{\mathcal{F}}^n$ represents the amount of w^n swept by each solid boundary \mathcal{F} present in the cell during the time step. The solid boundary \mathcal{F} is the largest subsegment of the solid boundary which is contained in one single cell (not necessarily the same) at times $n\Delta t$ and $(n+1)\Delta t$. The precise definition of \mathcal{F} and the expression of $\Delta w_{\mathcal{F}}^n$ will be given in section 4.3.

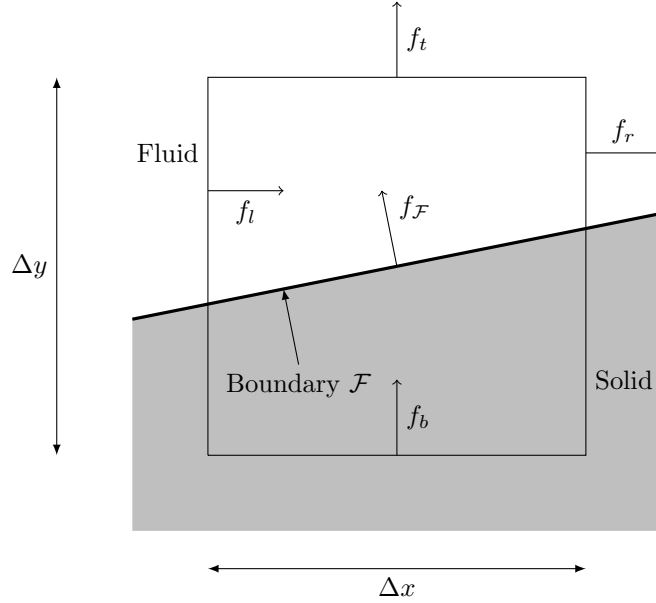


Figure 1: Physical fluxes in a cut cell

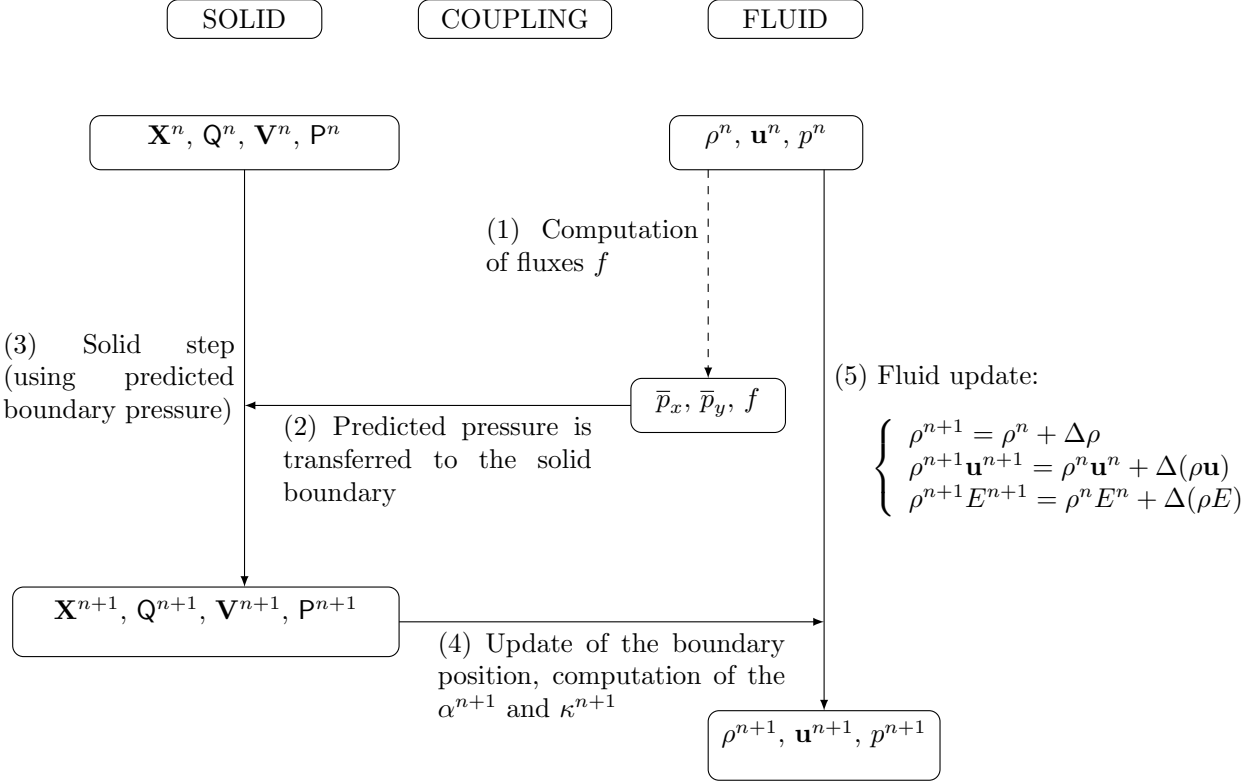
4 Coupling algorithm

Since the Discrete Element method is computationally expensive, the coupling algorithm should be explicit in order to avoid costly iterative procedures. In fact, the CFL condition of the explicit time-scheme gives the appropriate criterion for the capture of the high-frequency eigenmodes involved in the solid body fast dynamics. Moreover, as it is well known, explicit methods are more robust for impact problems. We choose the following general structure of the algorithm, which can be traced back to Noh [39]:

- The position of the solid and the density, velocity and pressure of the fluid are known at time t
- The fluid exerts a pressure force on the solid boundaries: knowing the total forces applied on the solid, the position of the solid is advanced to time $t + \Delta t$
- The density, velocity and pressure of the fluid are then computed at time $t + \Delta t$. This step takes into account the new position and velocity of the solid boundary, as well as the work of the forces of pressure on the boundary during the time step.

The choice of the coupling algorithm is guided by the conservation of the global momentum and energy of the system and the conservation of constant flows (see section 6).

At the beginning of a time step, at time $n\Delta t$, the position and rotation of the solid particle $(\mathbf{X}^n, \mathbf{Q}^n)$, the velocity and angular velocity of the solid particle $(\mathbf{V}^n, \mathbf{\Omega}^n)$ and the fluid state \mathbf{w}^n are known. We choose the following general architecture for the algorithm:



Steps (1) to (5) of the algorithm are computed successively, and are detailed in the following subsections.

4.1 Computation of fluid fluxes and of the boundary pressure (steps (1) and (2))

Step (1) is a precomputation of fluxes without considering the presence of a solid boundary. As said above, the fluxes are computed in every cell using the OSMP11 scheme. However, we emphasize that the coupling algorithm does not depend on the choice of the numerical scheme. The fluxes are then stored for later use in step (5).

The other aim of this step is the computation of mean pressures in each cut-cell during the time-step in each direction \bar{p}_x and \bar{p}_y . These pressures, transferred to the solid boundary in step (2), account for the forces exerted by the fluid on the solid during the time-step. The same mean pressures will be used in step (5) to compute the momentum and energy exchanged between the solid and the fluid. In this way, the choice of \bar{p}_x and \bar{p}_y has no effect on the conservation of fluid mass, momentum or energy of the system. On the contrary it is a key ingredient for the exact conservation of constant flows (see section 6). The explicit structure of our solid and fluid methods allows several possibilities for the choice of boundary pressures while maintaining the stability of the coupling algorithm. This is unusual in fluid-structure interaction.

The Strang directional splitting algorithm [47] is originally formulated as follows:

$$\mathbf{w}_j^{(n+1)} = L_x \left(\frac{\Delta t}{2} \right) L_y (\Delta t) L_x \left(\frac{\Delta t}{2} \right) \mathbf{w}_j^n,$$

where $L_x(\Delta t)$ and $L_y(\Delta t)$ are finite-difference approximation operators for the integration by a time-step Δt in directions x and y respectively. Here, this splitting procedure is implemented in a simplified form:

$$\mathbf{w}_j^{(n+2)} = L_x(\Delta t) L_y(\Delta t) L_y(\Delta t) L_x(\Delta t) \mathbf{w}_j^n,$$

that recovers the symmetry of the solution every two time steps. In our case, L_x and L_y involve the computation of a flux in the x or y direction using the state of fluid \mathbf{w} of the cells. The mean pressures \bar{p}_x and \bar{p}_y are then the pressures in the cell used for the computation of the fluxes by L_x and L_y , respectively.

An analogous definition could be derived for other time integration methods, such as Runge-Kutta for instance. The directional splitting used for the fluid flux computation does not require the solid body displacement to be split in x and y components. It is applied here only to recover second-order accuracy of the fluxes.

4.2 Computation of the solid step (step (3))

Step (3) consists mainly in the application of the time integration scheme for the rigid body motion described in section 2.1. The essential difference with an uncoupled version lies in the integration of boundary pressure forces. As we consider an explicit coupling, the only boundary pressures available are \bar{p}_x and \bar{p}_y .

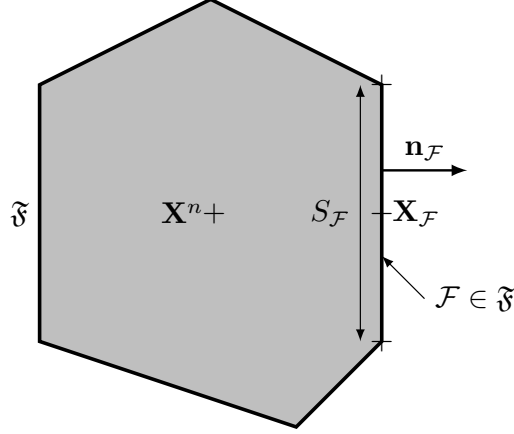


Figure 2: Geometric description of the particles

The solid is assumed to be polygonal (in two space dimensions) as described in Figure 2. We denote by \mathfrak{F} the list of all faces of the solid in contact with fluid. For every face $\mathcal{F} \in \mathfrak{F}$, the position of the center of the face is given by vector $\mathbf{X}_{\mathcal{F}}$, and we denote by $S_{\mathcal{F}}$ its surface and $\mathbf{n}_{\mathcal{F}}$ its normal vector (oriented from the solid to the fluid). The fluid pressure force $\mathbf{F}_{\mathcal{F}}$ exerted on face $\mathcal{F} \in \mathfrak{F}$ is written as:

$$\mathbf{F}_{\mathcal{F}} \cdot \mathbf{e}_x = -\bar{p}_x S_{\mathcal{F}} n_{\mathcal{F}}^x \quad (16)$$

$$\mathbf{F}_{\mathcal{F}} \cdot \mathbf{e}_y = -\bar{p}_y S_{\mathcal{F}} n_{\mathcal{F}}^y \quad (17)$$

The total fluid pressure force \mathbf{F}_f^n is the sum of the contributions on each face:

$$\mathbf{F}_f^n = \sum_{\mathcal{F} \in \mathfrak{F}} \mathbf{F}_{\mathcal{F}} \quad (18)$$

The fluid pressure torque \mathcal{M}_f^n is the sum of the torques of the pressure forces at the center of mass of the solid body:

$$\mathcal{M}_f^n = \sum_{\mathcal{F} \in \mathfrak{F}} \mathbf{F}_{\mathcal{F}} \wedge (\mathbf{X}^n - \mathbf{X}_{\mathcal{F}})$$

The solid time-step is written as in equations (1) to (8), with the only difference that the fluid pressure force and torque are taken constant during the whole time-step, equal to \mathbf{F}_f^n and \mathcal{M}_f^n (including in equations (6) and (7)). The fact that \mathbf{F}_f^n , \mathcal{M}_f^n , $\mathbf{V}^{n+\frac{1}{2}}$ and $\mathbf{P}^{n+\frac{1}{2}}$ are constant during the time-step will be used in the conservation analysis in section 5.

4.3 Update of the boundary and of the volume fractions (step (4))

Several tasks are carried out in step (4). For each cell \mathcal{C} , the new solid volume fraction of the cell $\alpha_{\mathcal{C}}^{n+1}$ and new surface fractions $\kappa_{\mathcal{C}}^{n+1}$ are computed. In addition, for each solid boundary \mathcal{F} , the pressures \bar{p}_x and \bar{p}_y are stored and the swept quantities $\Delta w_{\mathcal{F}}^n$ used in (15) are evaluated.

In two dimensions, the solid boundary is polygonal, and we therefore only have to deal with plane boundaries \mathcal{F} . In order to simplify the computation of the average of \bar{p}_x and \bar{p}_y on \mathcal{F} , we also assume that

each boundary \mathcal{F} is contained only in one cell at time $n\Delta t$. The computation of the contribution of $\Delta w_{\mathcal{F}}^n$ to each cell is also easier if \mathcal{F} is entirely in the cell at time $(n+1)\Delta t$. We denote $\Phi_n(\mathcal{F})$ the position of boundary \mathcal{F} at time $n\Delta t$. We choose to define \mathcal{F} as the largest subsegment of the boundary polygon such that $\Phi_n(\mathcal{F})$ is contained in cell \mathcal{C}_n at time $n\Delta t$ and $\Phi_{n+1}(\mathcal{F})$ is contained in cell \mathcal{C}_{n+1} at time $(n+1)\Delta t$ (see Figure 3). The two cells need not be necessarily different. \mathcal{F} may contain one or both vertices of the polygonal boundary at its ends, but we assume that each \mathcal{F} is contained in one single polygonal face. At each new time step, the polygonal boundary is subdivided into a new set of plane boundaries \mathcal{F} . Each newly computed boundary $\mathcal{F} \in \mathfrak{F}$ stores every variable necessary for the coupling: the surface $S_{\mathcal{F}}$ and the normal vector $\mathbf{n}_{\mathcal{F}}$ of $\Phi_n(\mathcal{F})$, the center of mass $\mathbf{X}_{\mathcal{F}}$ of \mathcal{F} , and we define $\mathbf{X}_{\mathcal{F}}^0 = \Phi_0(\mathbf{X}_{\mathcal{F}})$. The boundary also stores the pressures \bar{p}_x and \bar{p}_y in the cell occupied by $\Phi_n(\mathcal{F})$, and the velocity of the center of the boundary at time $(n + \frac{1}{2})\Delta t$, $\mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}}$, computed as:

$$\mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}} = \mathbf{V}^{n+\frac{1}{2}} + \mathbf{P}^{n+\frac{1}{2}} \mathbf{D}^{-1} \cdot (\mathbf{X}^0 - \mathbf{X}_{\mathcal{F}}^0) \quad (19)$$

The swept quantities $\Delta w_{\mathcal{F}}^n$ are computed as the integral of w^n in the quadrangle bounded by $\Phi_n(\mathcal{F})$ and $\Phi_{n+1}(\mathcal{F})$ (see Fig. 3). The condition

$$\sum_{\mathcal{F}} \Delta w_{\mathcal{F}}^n = \sum_{\mathcal{C}} (\alpha_{\mathcal{C}}^{n+1} - \alpha_{\mathcal{C}}^n) w_{\mathcal{C}}^n \quad (20)$$

is then automatically satisfied as the set of such quadrangles is a partition of the volume swept by the solid during the time step.

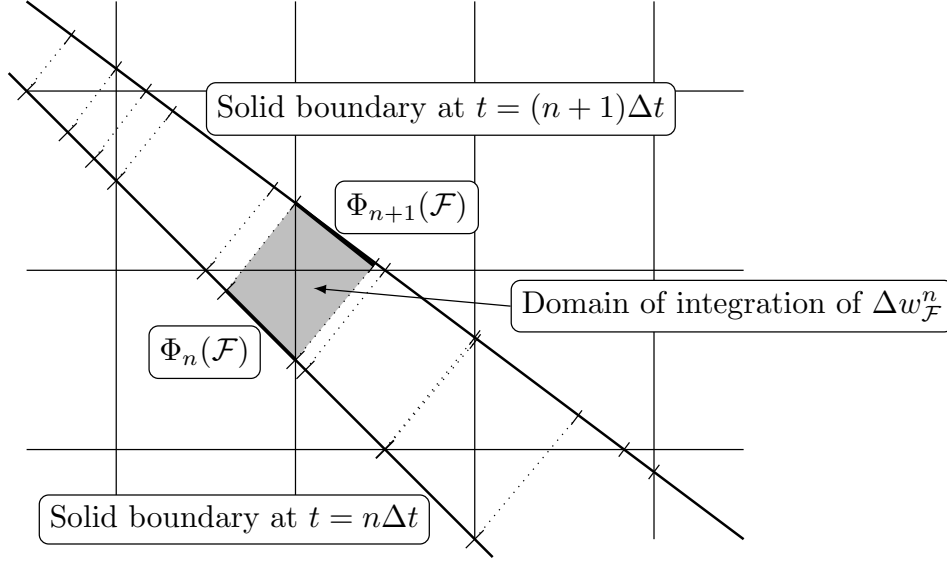


Figure 3: Update of the boundary and computation of the $\Delta w_{\mathcal{F}}^n$

The computation of $\alpha_{\mathcal{C}}^{n+1}$ and $\kappa_{\mathcal{C}}^{n+1}$ involves the intersection of planes with rectangles, and can be carried out geometrically. The application of the divergence theorem to cell \mathcal{C} shows that the following relations are satisfied:

$$\kappa_{\mathcal{C}l}^n = \kappa_{\mathcal{C}r}^n + \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta y} n_{\mathcal{F}}^x \quad (21)$$

$$\kappa_{\mathcal{C}b}^n = \kappa_{\mathcal{C}t}^n + \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta x} n_{\mathcal{F}}^y \quad (22)$$

These conditions correspond to the geometric conservation laws (GCL) in ALE methods [32], and will be used in the analysis of the consistency of our method.

4.4 Modification of fluxes (step (5))

Step (5) of the algorithm consists mainly in the computation of the final values $w_{\mathcal{C}}^{n+1}$ in each cell, using a fully discrete expression of Eq. (15). It is the only part of the algorithm where the fluid “sees” the

presence of a solid. The explicit fluid fluxes were pre-computed in step (1) on the Cartesian regular grid, and the modification of fluxes aims at conserving the mass of fluid and balancing the momentum and energy transferred to the solid during the time-step.

The exchange term in (15) can be written as

$$X_{\mathcal{F}} = \sum_{\mathcal{F} \in \mathcal{C}} S_{\mathcal{F}} f_{\mathcal{F}},$$

where $f_{\mathcal{F}}$ is the fluid flux at the solid boundary \mathcal{F} (see fig. 1), that are approximated as:

$$f_{\mathcal{F}} = \frac{1}{S_{\mathcal{F}}} \left(0, \int_{\mathcal{F}} \bar{p}_x n_{\mathcal{F}}^x, \int_{\mathcal{F}} \bar{p}_y n_{\mathcal{F}}^y, \mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}} \cdot \int_{\mathcal{F}} \begin{pmatrix} \bar{p}_x n_{\mathcal{F}}^x \\ \bar{p}_y n_{\mathcal{F}}^y \end{pmatrix} \right)^T \quad (23)$$

Here $\mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}}$ is the velocity of the center of the boundary and is defined in (19).

Using the fluid fluxes given by the OSMP11 scheme, we finally compute the time increment $\Delta w_{\mathcal{C}}$ from the following fully discrete version of equation (15):

$$(1 - \alpha_{\mathcal{C}}^{n+1}) \Delta w_{\mathcal{C}} = \Delta t \left(\frac{1 - \kappa_{cl}^{n+1}}{\Delta x} f_{cl} - \frac{1 - \kappa_{cr}^{n+1}}{\Delta x} f_{cr} + \frac{1 - \kappa_{cb}^{n+1}}{\Delta y} f_{cb} - \frac{1 - \kappa_{ct}^{n+1}}{\Delta y} f_{ct} \right) + \frac{\Delta t}{\Delta x \Delta y} \sum_{\mathcal{F} \in \mathcal{C}} S_{\mathcal{F}} f_{\mathcal{F}} + \sum_{\mathcal{F} \in \mathcal{C}} \Delta w_{\mathcal{F}}^n \quad (24)$$

The value of $w_{\mathcal{C}}$ is then updated in every cell: $w_{\mathcal{C}}^{n+1} = w_{\mathcal{C}}^n + \Delta w_{\mathcal{C}}$.

A main difference with [14] lies in the time integration of cell-face apertures $(1 - \kappa_{\mathcal{C}})$. Falcovitz *et al.* [14] use time-averaged cell-face apertures over the time step (at time $(n + \frac{1}{2})\Delta t$), ensuring consistency (in the sense that the uniform motion of a solid-fluid system is exactly preserved). In fact, a key ingredient in the consistency proof is the fact that conditions (21) and (22) are checked exactly. In [14], the consistent choice of time-averaged cell-face apertures $\kappa^{n+\frac{1}{2}}$ and solid surfaces in the fluid cell $\tilde{S}^{n+\frac{1}{2}}$ allows to check these conditions.

Here we instead take κ^{n+1} and recover consistency using the solid surface \tilde{S}^{n+1} present in the fluid cell at time $(n + 1)\Delta t$. This result is proved in section 6. Note that $\tilde{S}_{\mathcal{C}}^{n+1} = \sum_{\Phi_{n+1}(\mathcal{F}) \in \mathcal{C}} S_{\mathcal{F}}^n$ as the solid is undeformable. This choice is motivated by the fact that the computation of time averaged $\kappa^{n+\frac{1}{2}}$ and $\tilde{S}^{n+\frac{1}{2}}$ is already complex in two dimensions and might become intractable in three dimensions. In addition, it requires an implicit resolution of $\tilde{S}^{n+\frac{1}{2}}$ in order to preserve the energy of the system. The choice of κ^{n+1} theoretically reduces the accuracy of the method in cut-cells. However, the accuracy of our numerical results did not advocate the use of the time-averaged $\kappa^{n+\frac{1}{2}}$ and the related added complexity in the algorithm.

In order to avoid the classical restriction of the time-step due to vanishing volumes:

$$\Delta t \leq \frac{(1 - \alpha_{\mathcal{C}}) \min(\Delta x, \Delta y)}{\|\mathbf{u}\| + c},$$

where c is the local speed of sound, we resort to the mixing of small cut cells with their neighbors to prevent instabilities.

4.5 Conservative mixing of small cut cells

Two main methods have been developed to ensure the stability of conservative Embedded Boundary methods. A first method consists in computing a reference state using nonconservative interpolations, modified by redistributing the conservation error on neighbouring cells [43, 12, 36]. A second method is to compute a fully conservative state using a formula similar to (24). For stability reasons, small cells are merged with neighboring cells using a conservative procedure (originating from Glimm's idea [22]). We choose this second class of method, as [25] and [14].

Target cells need to be defined for small cells to be merged with them. [25] defines an equivalent normal vector to the boundary in the cell and mixes the cells preferentially in that direction. [14] rather merges newly exposed or newly covered cells with full neighbours having a face in common. In order to deal with cells occupied by several boundaries (impact of two solids), we cannot define a normal vector

in every cell and we choose to improve the strategy applied in [14]. We define small cells as $\alpha_C > 0.5$. For mixing two cells C and C_T , so they have equal final value w , the following quantities are exchanged:

$$\begin{aligned} M_{CC_T} &= \frac{\alpha_{C_T}}{\alpha_C + \alpha_{C_T}} (w_{C_T} - w_C) \\ M_{C_TC} &= \frac{\alpha_C}{\alpha_C + \alpha_{C_T}} (w_C - w_{C_T}) \end{aligned}$$

and it is easy to check that $w_C + M_{CC_T} = w_{C_T} + M_{C_TC}$. In the two dimensional case, we select the target cell C_T as the fully-fluid cell ($\alpha_{C_T} = 0$) nearest to C , such that the path between the two cells does not cross a solid boundary. A recursive subroutine finds such a target cell in a small number of iterations, without any restriction on the geometry of the fluid domain.

5 Analysis of the conservation of mass, momentum and energy

In this section, we analyze the conservation properties of the coupling algorithm. These properties are verified for periodic boundary conditions or for an infinite domain.

5.1 Integration on the fluid domain

Integrating w on the fluid domain Ω_f^{n+1} at time $(n+1)\Delta t$, we obtain using (24) and the cancellation of fluxes on each cell face:

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \int_{\Omega_f^{n+1}} w^{n+1} &= \sum_C (1 - \alpha_C^{n+1}) w_C^n + \sum_C (1 - \alpha_C^{n+1}) \Delta w_C^n \\ &= \sum_C (1 - \alpha_C^{n+1}) w_C^n + \sum_{\mathcal{F}} \frac{\Delta t S_{\mathcal{F}}}{\Delta x \Delta y} f_{\mathcal{F}} + \sum_{\mathcal{F}} \Delta w_{\mathcal{F}}^n \end{aligned}$$

Using (20) we finally get:

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \int_{\Omega_f^{n+1}} w^{n+1} &= \sum_C (1 - \alpha_C^n) w_C^n + \sum_{\mathcal{F}} \frac{\Delta t S_{\mathcal{F}}}{\Delta x \Delta y} f_{\mathcal{F}} \\ &= \frac{1}{\Delta x \Delta y} \int_{\Omega_f^n} w^n + \frac{\Delta t}{\Delta x \Delta y} \sum_{\mathcal{F}} S_{\mathcal{F}} f_{\mathcal{F}} \end{aligned} \quad (25)$$

the expression of $f_{\mathcal{F}}$ being given in Eq. 23.

The first component of system (25) expresses the fluid mass conservation. In order to proceed with the analysis of momentum and energy conservation, let us now turn to the solid part.

5.2 Solid conservation balance

Since the solid is treated using a Lagrangian method, the conservation of solid mass is straightforward. The fluid pressure force applied on the solid during the time step is given by (16), (17) and (18). Let us consider a solid boundary $\mathcal{F} \in \mathfrak{F}$, and denote by $\Delta \mathcal{P}_{\mathcal{F}}$ the solid momentum variation induced by the pressure forces on \mathcal{F} , and $\Delta \mathcal{E}_{\mathcal{F}}$ the corresponding energy variation. Recalling that the pressure forces are kept constant during the time step, the balance of momentum and energy is given by:

$$\begin{aligned} \Delta \mathcal{P}_{\mathcal{F}} &= \Delta t \mathbf{F}_{\mathcal{F}} \\ \Delta \mathcal{E}_{\mathcal{F}} &= \Delta t \mathbf{F}_{\mathcal{F}} \cdot \left(\frac{1}{S_{\mathcal{F}}} \int_{\mathcal{F}} \mathbf{V}^{n+\frac{1}{2}}(\mathbf{x}) d\mathbf{x} \right) = \Delta t \mathbf{F}_{\mathcal{F}} \cdot \mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}} \end{aligned}$$

Finally, using the expression of forces $\mathbf{F}_{\mathcal{F}}$, we obtain:

$$\begin{aligned} \Delta \mathcal{P}_{\mathcal{F}}^x &= -\Delta t \int_{\mathcal{F}} \bar{p}_x n_{\mathcal{F}}^x \\ \Delta \mathcal{P}_{\mathcal{F}}^y &= -\Delta t \int_{\mathcal{F}} \bar{p}_y n_{\mathcal{F}}^y \\ \Delta \mathcal{E}_{\mathcal{F}} &= -\Delta t \mathbf{V}_{\mathcal{F}}^{n+\frac{1}{2}} \cdot \int_{\mathcal{F}} \begin{pmatrix} \bar{p}_x n_{\mathcal{F}}^x \\ \bar{p}_y n_{\mathcal{F}}^y \end{pmatrix} \end{aligned}$$

Comparing with section 5.1, the balance of momentum and energy in the fluid domain results in:

$$\begin{aligned}\int_{\Omega_f^{n+1}} \rho^{n+1} \mathbf{u}^{n+1} + \sum_{\mathcal{F}} \Delta \mathcal{P}_{\mathcal{F}} &= \int_{\Omega_f^n} \rho^n \mathbf{u}^n \\ \int_{\Omega_f^{n+1}} \rho^{n+1} E^{n+1} + \sum_{\mathcal{F}} \Delta \mathcal{E}_{\mathcal{F}} &= \int_{\Omega_f^n} \rho^n E^n\end{aligned}$$

This demonstrates the conservation of momentum and energy for the coupled system.

6 Conservation of constant flows

In this section we analyze the consistency of the coupling method, in the sense defined in [14], meaning exact conservation of uniform flows by the coupling algorithm. Two cases are analyzed. The first one, also considered in [14], consists of a solid immersed in a fluid and moving at the same velocity. This property is called “consistency” in [14]. The second one, not considered before, demonstrates the correct representation of the slip boundary condition along walls. These simple cases have been a guide to design the algorithm, as the preservation of such flows is a basic criterion for the quality of the method.

In the whole section, we consider a constant fluid state: $\rho^n = \rho_0$, $u^n = u_0$, $v^n = v_0$ and $p^n = p_0$ everywhere. The fluxes f are such that $f_r = f_l = (\rho_0 u_0, \rho_0 u_0^2 + p_0, \rho_0 u_0 v_0, (\rho_0 e_0 + p_0) u_0)^T$ and $f_t = f_b = (\rho_0 v_0, \rho_0 u_0 v_0, \rho_0 v_0^2 + p_0, (\rho_0 e_0 + p_0) v_0)^T$. In this case, the efficient pressures on the boundary of the solid are $\bar{p}_x = \bar{p}_y = p_0$.

6.1 Steady constant flow with moving boundaries

We consider an arbitrarily shaped rigid body, moving at constant velocity with no rotation, immersed in a uniform fluid flowing at the same velocity.

The solid is a closed set, and we denote by Ω_s^n the solid domain at initial time. We have:

$$\sum_{\mathcal{F}} S_{\mathcal{F}} \mathbf{n}_{\mathcal{F}} = \oint_{\partial \Omega_s^n} \mathbf{n} dS = \mathbf{0}$$

Using (16) and (17), we obtain:

$$\sum_{\mathcal{F}} \mathbf{F}_{\mathcal{F}} = - \sum_{\mathcal{F}} p_0 S_{\mathcal{F}} \mathbf{n}_{\mathcal{F}} = \mathbf{0}$$

This induces:

$$\mathbf{V}_i^{n+1} = \mathbf{V}_i^{n+\frac{1}{2}} = \mathbf{V}_i^n = (u_0, v_0)^T, \quad \mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \Delta t (u_0, v_0)^T$$

In the same way,

$$\sum_{\mathcal{F}} \mathcal{M}_{\mathcal{F}} = - \sum_{\mathcal{F}} p_0 S_{\mathcal{F}} \mathbf{n}_{\mathcal{F}} \wedge (\mathbf{X}_i^n - \mathbf{X}_{\mathcal{F}}) = p_0 \oint_{\partial \Omega_s} (\mathbf{X}_i^n - \mathbf{X}) \wedge \mathbf{n} dS = \mathbf{0}$$

The volume swept by boundary \mathcal{F} is $\Delta t S_{\mathcal{F}} (u_0, v_0)^T \cdot \mathbf{n}_{\mathcal{F}}$. Since the initial state is constant, $\Delta w_{\mathcal{F}}^n$ is given by:

$$\Delta w_{\mathcal{F}}^n = \frac{\Delta t S_{\mathcal{F}}}{\Delta x \Delta y} (u_0 n_{\mathcal{F}}^x + v_0 n_{\mathcal{F}}^y) w_0$$

In addition, as the solid translates without rotation, the normal vector $\mathbf{n}_{\mathcal{F}}$ to a boundary \mathcal{F} is constant in time. Using this property in equations (21) and (22), we easily conclude that $(1 - \alpha_C) \Delta w_C = 0$. Thus $w^{n+1} = w^n$, showing that the constant flow is left unchanged by the coupling algorithm and the mixing of small cells.

6.2 Free slip along a straight boundary

We consider an undeformable, fixed solid consisting in a semi-infinite half-space. The solid boundary is a straight planar boundary with a constant normal vector \mathbf{n} such that:

$$\mathbf{u}_0 \cdot \mathbf{n} = 0 \tag{26}$$

This initial state describes the free slip of the fluid along the straight boundary. In the inviscid case, no boundary layer should develop in the vicinity of the boundary. The conservation of such flows ensures that the boundary is not seen by the fluid as being artificially rough.

As the solid is fixed, α_C and κ_C remain constant over time and $\Delta w_{\mathcal{F}}^n$ is equal to zero. From equation (24), and using (21), (22) and (26), the components of Δw_C are calculated as:

$$\begin{aligned}(1 - \alpha_C)\Delta\rho_C &= -\Delta t \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta x \Delta y} \mathbf{n} \cdot \mathbf{u}_0 = 0 \\(1 - \alpha_C)\Delta(\rho u)_C &= -\Delta t \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta x \Delta y} ((\mathbf{n} \cdot \mathbf{u}_0)u_0 + p_0 n_x) + \sum_{\mathcal{F} \in \mathcal{C}} \frac{\Delta t S_{\mathcal{F}}}{\Delta x \Delta y} p_0 n_x = 0 \\(1 - \alpha_C)\Delta(\rho v)_C &= -\Delta t \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta x \Delta y} ((\mathbf{n} \cdot \mathbf{u}_0)v_0 + p_0 n_y) + \sum_{\mathcal{F} \in \mathcal{C}} \frac{\Delta t S_{\mathcal{F}}}{\Delta x \Delta y} p_0 n_y = 0 \\(1 - \alpha_C)\Delta(\rho E)_C &= -\Delta t \sum_{\mathcal{F} \in \mathcal{C}} \frac{S_{\mathcal{F}}}{\Delta x \Delta y} (\mathbf{n} \cdot \mathbf{u}_0)(\rho_0 e_0 + p_0) = 0\end{aligned}$$

This shows that the constant flow is preserved by step (5) of the algorithm. This result is not modified by the mixing procedure. We thus have shown the exact preservation of the free slip of the fluid along a straight boundary.

7 Numerical examples

In the following, we consider a perfect gas, with $\gamma = 1.4$. In all computations the CFL number was fixed equal to 0.5.

7.1 One-dimensional results

A piston of density 2 kg.m^{-3} and length 0.5 m is initially centered at $x = 2 \text{ m}$, in a one-dimensional, 7m -long tube, whose ends are connected by periodic boundary conditions which allow an easier comparison with ALE results. The gas initial pressure and density are equal to 10^6 Pa and 10 kg.m^{-3} for $x \leq 2\text{m}$ and $x \geq 5\text{m}$ and to 10^5 Pa and 1 kg.m^{-3} elsewhere. The system is initially at rest. The initial pressure difference between the two sides of the piston triggers its movement and the propagation of waves in the fluid regions (a rarefaction in the left region and a shock wave in the right one). Wave interactions then occur at later time. The fluid pressure at time $t = 0.003\text{s}$ is shown in Figure 4, and the trajectory of the solid is presented in Figure 5. The $x - t$ diagram over a longer time (0.01 s) is shown in Figure 6.

An ALE computation was done for comparison, using a uniform grid moving at the solid velocity. The solid position and velocity are updated using the same second-order Verlet scheme. We compared the numerical results obtained through the Embedded Boundary method on 100, 200, 400, 800, 1600, 3200, 6400 and 12800 points grids with a 51200-points ALE grid, considered as the reference solution. We observe a second-order convergence of the solid position (Figure 7) and a super-linear convergence of order 1.2 of the fluid pressure (Figure 8). The convergence rate is optimal for the solid (Verlet scheme is second-order accurate). The convergence rate for the fluid pressure is not optimal, due to the presence of discontinuities, but is not affected by the solid coupling.

7.2 Double Mach reflection

A Mach 10 planar shock wave reflects on a fixed 30° wedge, creating a Mach front, a reflected shock wave, and a contact discontinuity which develops into a jet along the solid boundary. This benchmark was first simulated on a Cartesian grid aligned with the solid boundary, using different finite volume methods [27, 48, 8]. Non-aligned grid methods were also tested on this benchmark, using Embedded Boundary methods [43, 7], non-conservative Immersed Boundary methods [21], h -box methods [24], and kinetic schemes [31]. The position of the tip of the jet is an important characteristic of the accuracy of the results. The comparison with grid-aligned results shows that it is better recovered by conservative methods than non-conservative methods [43, 7].

We have simulated the problem on a grid aligned with the wedge (aligned case, Figure 9) and on a grid aligned with the incident shock wave (non-aligned case, Figure 10). The two results are very similar,

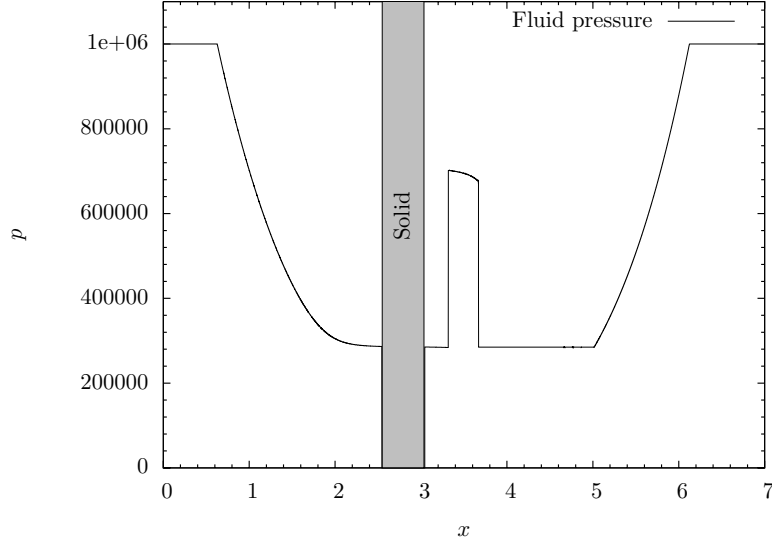


Figure 4: Pressure at time $t = 0.003$ s

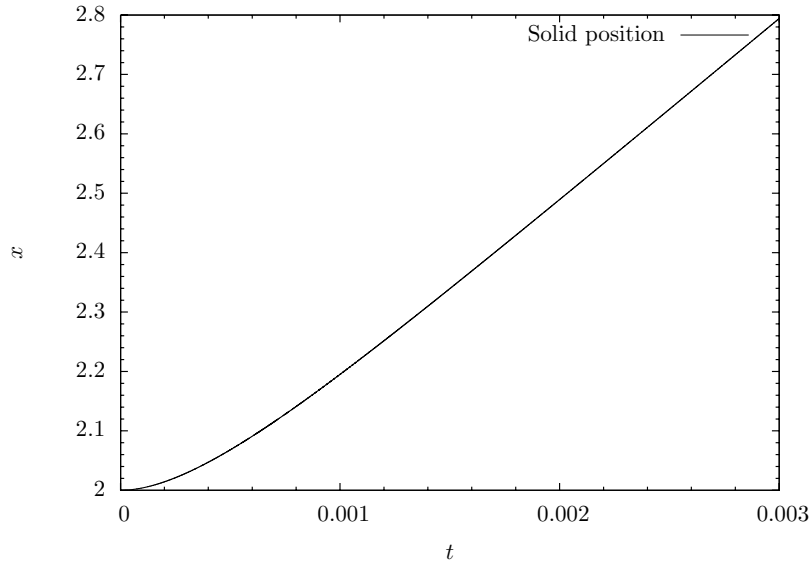


Figure 5: Time evolution of the solid position

and agree with [7, 43, 27, 8]. One can remark that all the features of the flow are captured at the correct position in the non-aligned case. The jet propagates along the wall without numerical friction due to the conservation of free slip along a straight boundary (section 6.2). In the principal Mach stem, the discontinuities are slightly more oscillatory than in the aligned case. This can be identified as a post-shock oscillation phenomenon to which Roe's scheme is especially prone (see, for instance, [28, 4]), and is not related to the coupling method. Nevertheless, the perturbations stay localized in the vicinity of discontinuities.

7.3 Lift-off of a cylinder

This moving body test case was first proposed in [14], using a conservative method. A rigid cylinder of density 7.6kg.m^{-3} and diameter 0.1 m, initially resting on the lower wall of a $1\text{m} \times 0.2\text{m}$ two-dimensional channel filled with air at standard conditions ($\rho = 1\text{ kg.m}^{-3}$, $p = 1\text{ Pa}$), is driven and lifted upwards by a

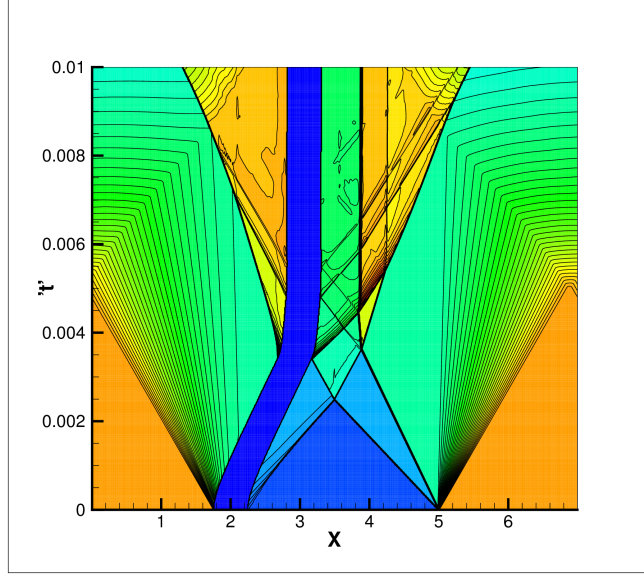


Figure 6: $x - t$ diagram (the position of the solid is in deep blue)

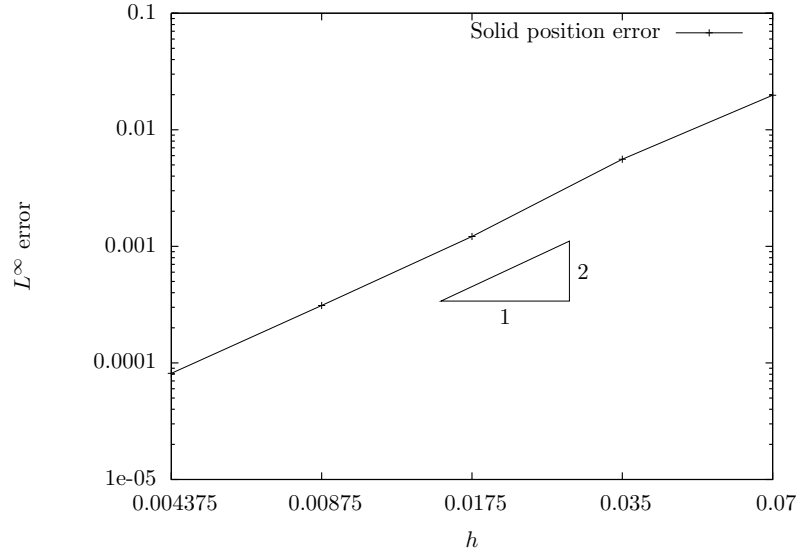


Figure 7: Convergence of the solid position L^∞ -error

Mach 3 shock wave. Gravity is not taken into account. The problem was simulated in [20, 3, 25, 46]. In Figure 11, we present our results on a uniform 1600×320 grid at times 0.14 s and 0.255 s. The cylinder was approximated by a polygon with 1240 faces.

Our results agree well on the position of the solid and of the shocks with Arienti et al. [3] and Hu et al. [25]. However, some differences should be noted. First of all, some reflected shock waves in our results seem to lag slightly behind their position in previous results. This difference might be caused by small differences in the final position of the solid. Hu et al. [25] also discuss the presence of a strong vortex under the cylinder in the results of Forrer and Berger [20]. They dismiss it as an effect of the space-time splitting scheme employed which affects the numerical dissipation. We also obtain this vortex, which does not disappear as we refine the mesh. We rather believe that this vortex is associated with a Kelvin-Helmholtz instability of the contact discontinuity present under the cylinder (Figure 12).

In Figures 13 and 14 we present convergence results on the final position of the center of mass of the cylinder, compared to those of Hu et al. [25]. We observe that our results exhibit a fast convergence

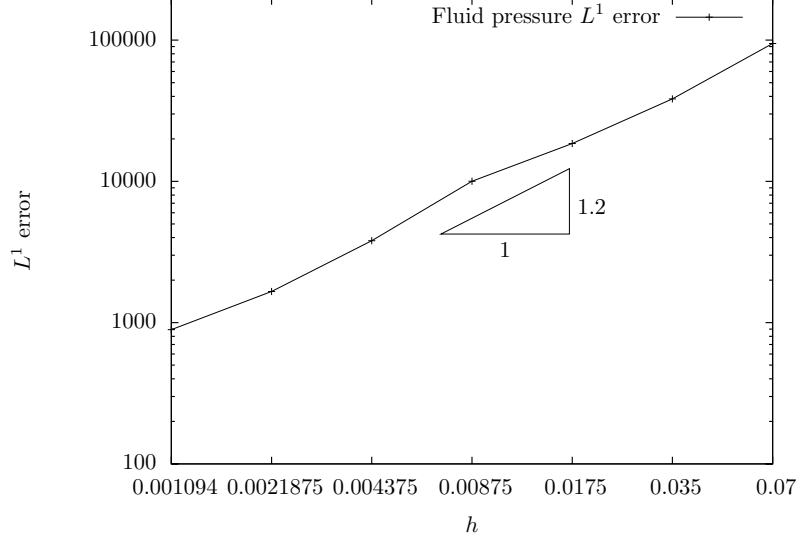


Figure 8: Convergence of the fluid pressure L^1 -error

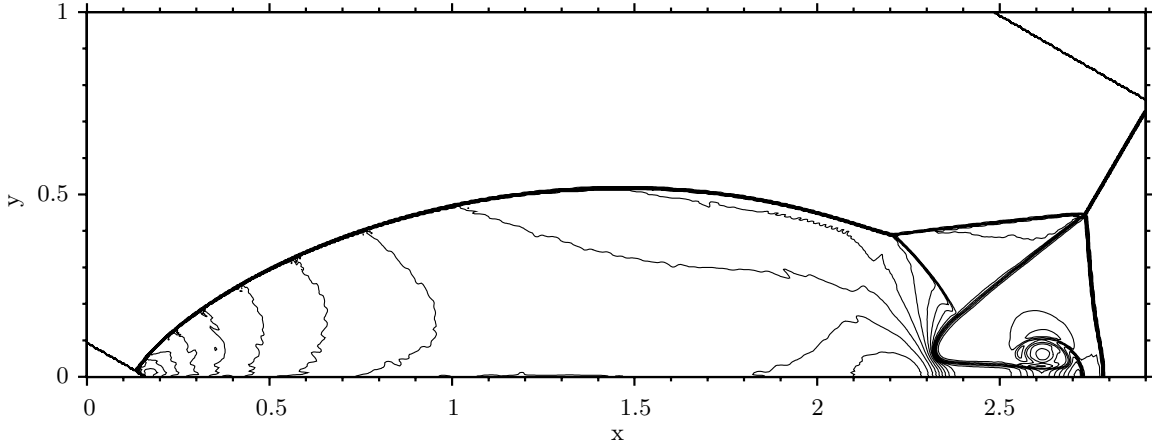


Figure 9: Aligned case: 30 contours of fluid density from 1.73 to 21, $\Delta x = \Delta y = 1/220$

process, which is not the case in [25]. Let us note that no exact solution exists for the final position of the cylinder. The final position we found is however in the same range as in [25]. The results also compare well with Arienti et al. [3]. The improvement lies in the combination of the conservative interface method [25] with a conservative coupling and a second-order time-scheme for the rigid body motion. For this difficult case, the maximal conservation relative errors due to coupling were bounded by $4 \cdot 10^{-6}$ over the whole simulation time, and no drift was observed.

In Figure 15 we present the relative computational cost of the coupling. The relative cost is defined as the ratio of the computational times dedicated to the coupling method and to the fluid and solid methods. In the rigid body case, the cost of the solid method is very low compared to that of the fluid method. As the coupling method is explicit and local, the computational cost is located on a manifold one dimension lower than the dimension of the whole space. In the two-dimensional case, the coupling is on a one-dimensional manifold. Indeed, we observe in Figure 15 that the relative cost of the coupling decreases as the grid is refined, with a slope of 0.5, and that the coupling cost remains lower than the fluid and solid costs, amounting to approximately 10–20% for the grids yielding sufficient accuracy.

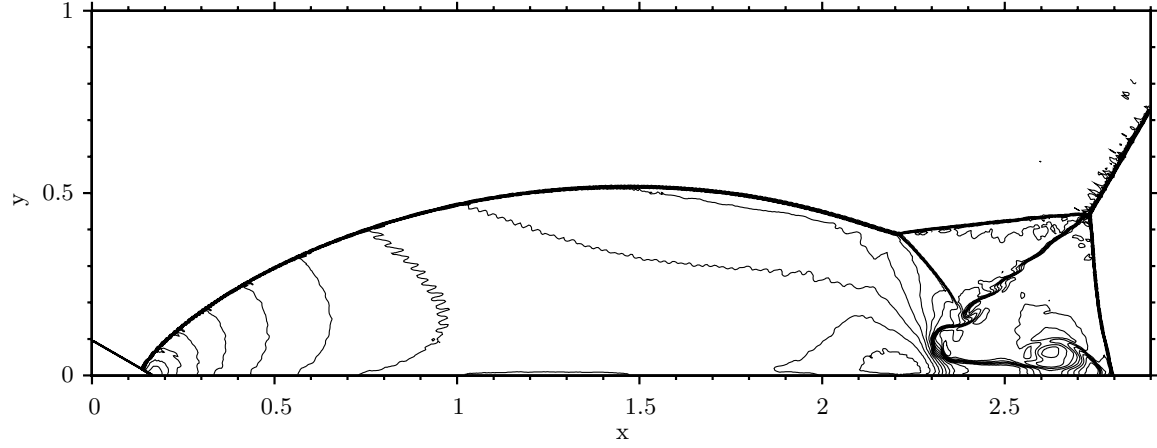


Figure 10: Non-aligned grid case: 30 contours of fluid density from 1.73 to 21, $\Delta x = \Delta y = 1/220$

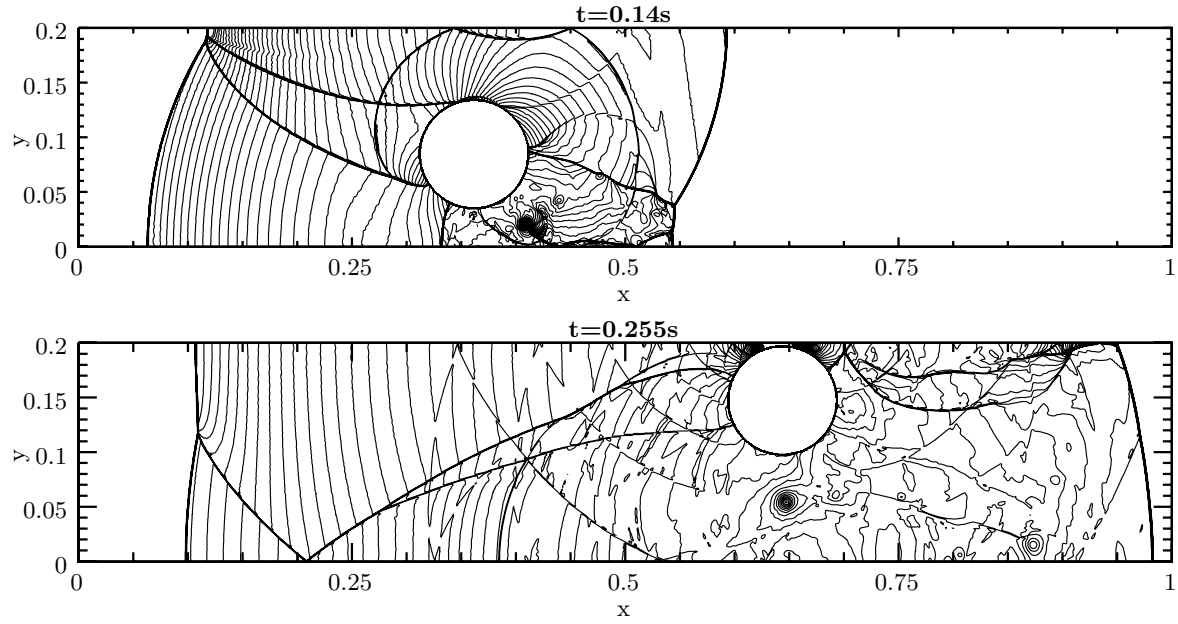


Figure 11: 60 contours of fluid pressure from 0 to 28 at different times, $\Delta x = \Delta y = 6.25 \times 10^{-4}$

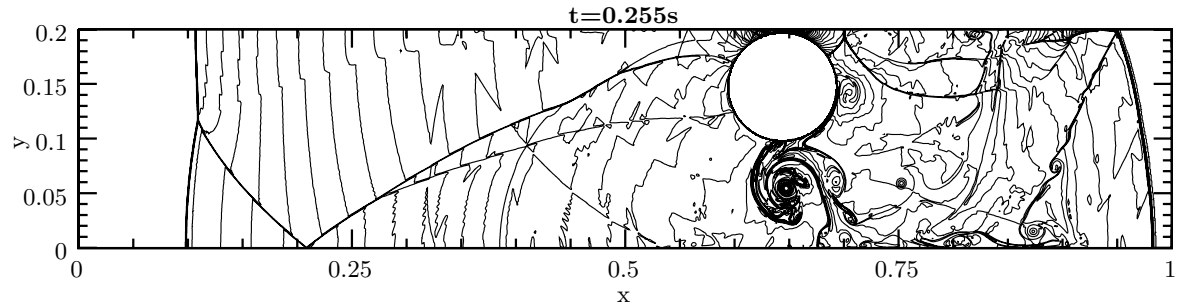


Figure 12: 60 contours of fluid density from 0 to 12 at final time, $\Delta x = \Delta y = 6.25 \times 10^{-4}$

7.4 Flapping doors

We propose this new fluid-structure interaction case as a demonstration of the robustness of our approach and as a first step towards fracture and impact simulations. The flapping doors case involves separating

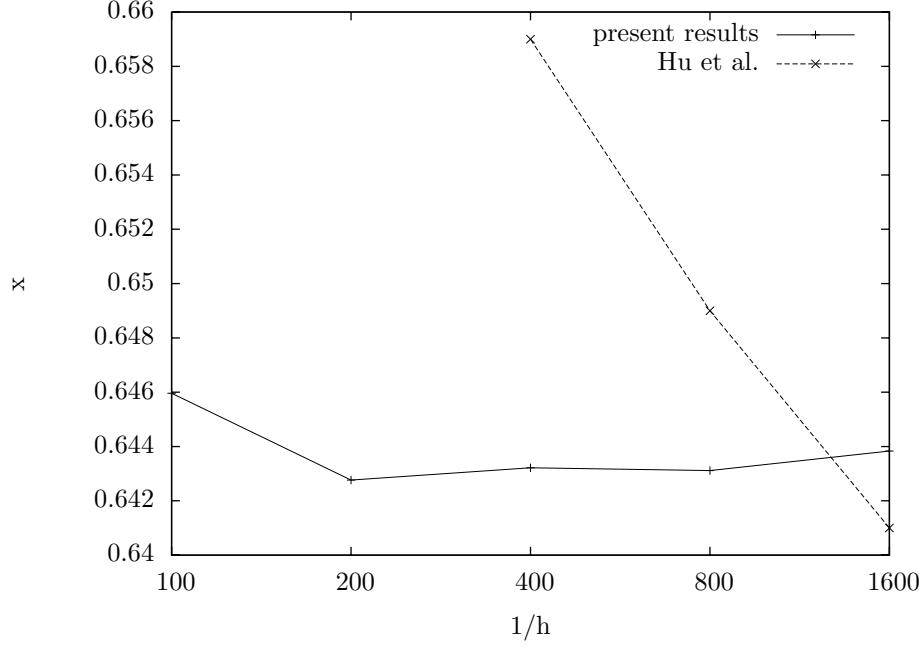


Figure 13: Comparison of the convergence of the horizontal position of the center of mass of the cylinder with Hu et al. [25]

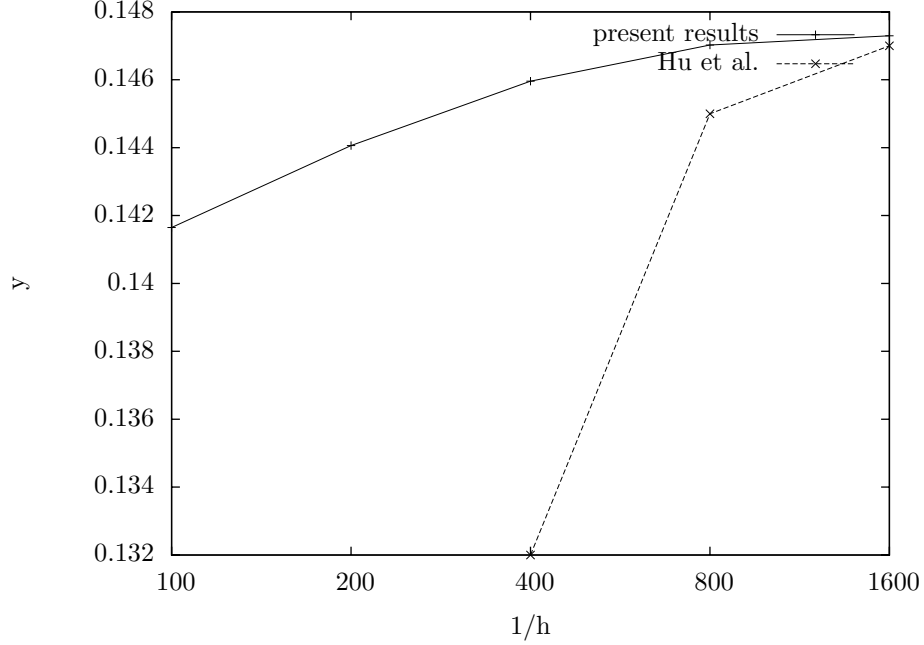


Figure 14: Comparison of the convergence of the vertical position of the center of mass of the cylinder with Hu et al. [25]

or closing solid boundaries, with cells including several moving boundaries. The algorithm is shown to be able to deal with such difficulties. Two doors initially close a canal and are impacted from the left by a Mach 3 shock. The canal consists of two fixed rigid walls, 2m long and 0.5m apart. Each door consists of a 0.2-m long and 0.05m-wide rectangle, completed at both ends by a half-circle of diameter 0.05m. The doors are respectively fixed on points (0.5, 0.025) and (0.5, 0.475) at the center of the half-circles. They can rotate freely around these points. The Mach 3 shock is initially located at $x = 0.43$ m. The density

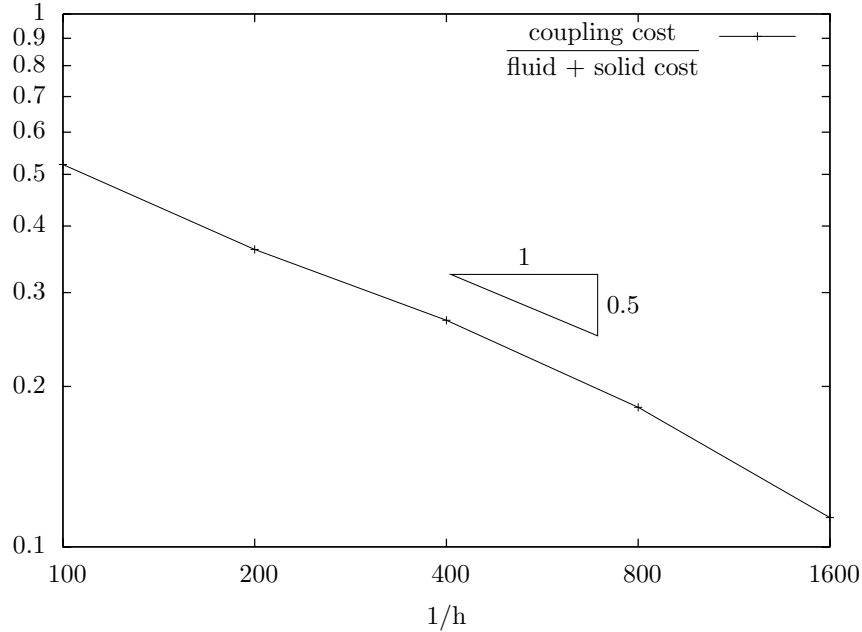


Figure 15: Ratio between the coupling computation cost and the fluid and solid costs

of the solid is 0.1 kg.m^{-3} and the pre- and post-shock state of the fluid are $(\rho, u, v, p) = (1, 0, 0, 1)$ and $(\rho, u, v, p) = (3.857, 2.6929, 0, 10.333)$. In Figure 16 we show the density field obtained using a 1600×400 grid, at times 0.125s, 0.25s, 0.375s and 0.5s. After the incident shock hits the doors, it reflects to the left and the doors open due to the high rise in pressure. The opening of the doors produces a jet preceded by a shock wave propagating to the right. Then complex interactions of waves occur, due to door movements and interaction with the walls. Kelvin-Helmholtz instabilities of contact lines can be observed at $t=0.5$ s. It is worth noting that symmetry of the flow about the centerline of the canal axis is remarkably well preserved by the coupling method.

As the doors remain tangent to the canal walls during their rotation, the fluid cannot pass between the wall and a door at its hinge. When the doors approach the walls at maximum rotation, the fluid is compressed, and eventually pushes them back. This is observed at time $t = 0.2162$ s and $t = 0.486$ s in Figure 17, which presents the time evolution of the doors rotation angle. In the first case, the distance between each door's straight boundary and the wall is less than 0.002 m, while the size of a fluid cell is $\Delta x = 0.00125$ m. The method is able to deal with the fact that most cells along the wall are cut by the moving boundary and contain several moving boundaries. Treating this test case with an ALE method would require several remeshings in the course of the simulation, especially in the initial separation of the tangent door tips, and when the doors approach the walls.

8 Conclusion

We have presented a new coupling algorithm between a compressible fluid flows and a rigid body using an Embedded Boundary method. This explicit algorithm has the advantage of preserving the usual CFL stability condition: the time-step can be taken as the minimum of the full cell size fluid and solid time-steps. The combination of the Embedded Boundary method for the fictitious fluid domain and of the coupling strategy ensures the conservation of fluid mass and the balance of momentum and energy between fluid and solid. In addition, the exact conservation of the two constant states described in section 6 gives good insight on the consistency of the method: we prove the conservation of a constant flow in which a solid moves at the same velocity and the fact that the treatment at the boundary introduces no spurious roughness or boundary layers. The numerical examples suggest the second-order convergence of the solid position and the super-linear convergence of the fluid state in L^1 norm, while our results on two-dimensional benchmarks agree very well with body-fitted methods and improve Immersed

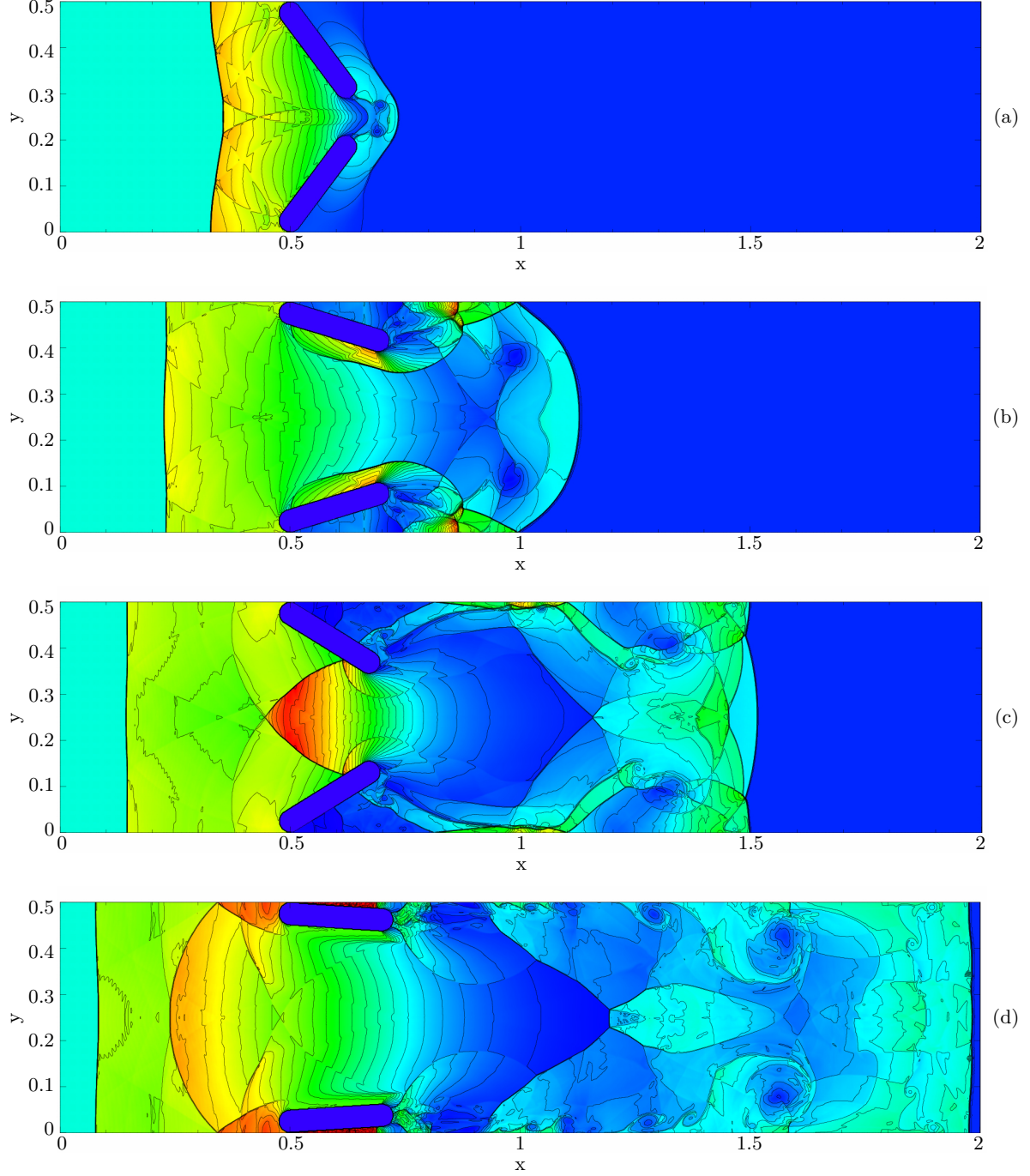


Figure 16: Density contours at times $t = 0.125s$ (a), $t = 0.25s$ (b), $t = 0.375s$ (c) and $t = 0.5s$ (d)

Boundary results. We are also capable of dealing with solid boundaries moving close to each other, which is promising for impact simulations. The method is computationally efficient, as the coupling adds an integration on a space one dimension smaller than the fluid and solid computation spaces. The present method is therefore perfectly liable to be extended to a deformable solid, and was designed to extend naturally in three space dimensions. The remaining difficulty is the ability to define and track the solid boundary surrounding a Discrete Element assembly.

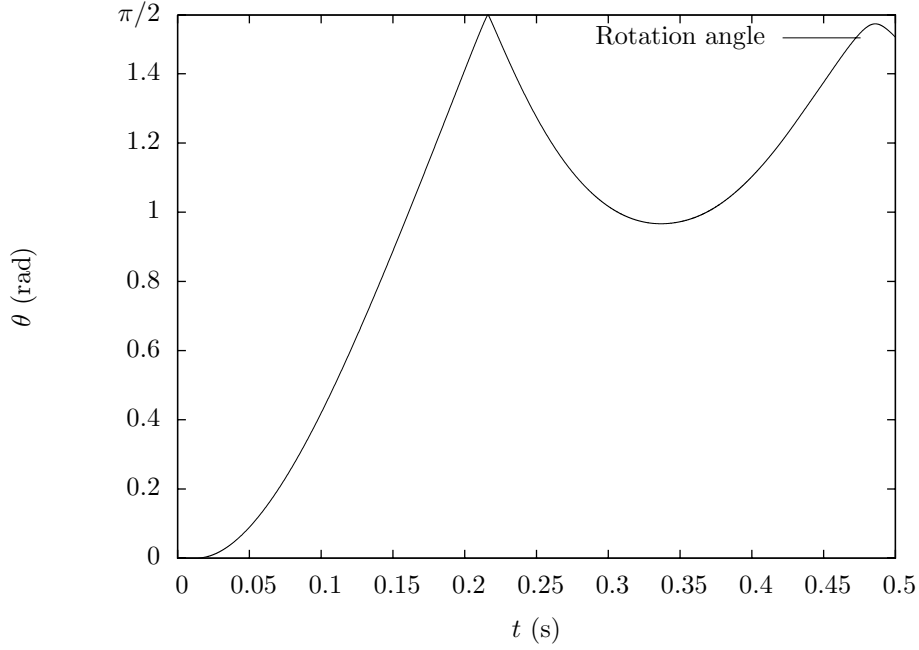


Figure 17: Time evolution of the rotation of a door

References

- [1] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *J. Comput. Phys.*, 125:150–160, 1996.
- [2] N. Aquelet, M. Souli, and L. Olovsson. Euler-lagrange coupling with damping effects : Application to slamming problems. *Comput. Methods Appl. Mech. Engrg.*, 195(1-3):110–132, 2006.
- [3] R. Arienti, P. Hung, E. Morano, and J.E. Shepherd. A level set approach to Eulerian-Lagrangian coupling. *J. Comput. Phys.*, 185:213–251, 2003.
- [4] M. Arora and P.L. Roe. On postshock oscillations due to shock capturing schemes in unsteady flows. *J. Comput. Phys.*, 130:25–40, 1997.
- [5] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large mach number flows in the presence of obstacles. *J. Comp. Fluid*, 38(3):703–714, 2009.
- [6] J.-I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *J. Comput. Phys.*, 224(2):757–784, 2007.
- [7] P. Colella, D.T. Graves, B.J. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *J. Comput. Phys.*, 211:347–366, 2006.
- [8] V. Daru and C. Tenaud. High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations. *J. Comput. Phys.*, 193(2):563–594, 2004.
- [9] V. Daru and C. Tenaud. Numerical simulation of the viscous shock tube problem by using a high resolution monotonicity-preserving scheme. *Comput. Fluids*, 38(3):664–676, 2009.
- [10] M. de Tullio and G. Iaccarino. Immersed boundary technique for compressible flow simulations on semi-structured meshes. CTR Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford Univ., 2005.
- [11] J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary Lagrangian Eulerian finite element method for transient dynamic fluid-structure interactions. *Comput. Methods Appl. Mech. Engrg*, 33:689–723, 1982.

- [12] Z. Dragojlovic, F. Najmabadi, and M. Day. An embedded boundary method for viscous, conducting compressible flow. *J. Comput. Phys.*, 216(1):37–51, 2006.
- [13] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J. Comput. Phys.*, 161(1):35–60, 2000.
- [14] J. Falcovitz, G. Alfandary, and G. Hanoch. A two-dimensional conservation laws scheme for compressible flows with moving boundaries. *J. Comput. Phys.*, 138:83–102, 1997.
- [15] C. Farhat, K.G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 195:1973–2001, 2006.
- [16] N. Favrie, S.L. Gavriluk, and R. Saurel. Solid-fluid diffuse interface model in cases of extreme deformation. *J. Comput. Phys.*, 228:6037–6077, 2009.
- [17] R.P. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the Ghost Fluid method. *J. Comput. Phys.*, 175:200–224, 2002.
- [18] R.P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid method). *J. Comput. Phys.*, 152:457–492, 1999.
- [19] M.A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *Int. J. Numer. Meth. Engrg.*, 69:794–821, 2007.
- [20] H. Forrer and M. Berger. *Flow simulation on Cartesian grids involving complex moving geometries flows*, volume 129 of *Int. Ser. Numer. Math.* Birkhäuser, 1998.
- [21] H. Forrer and R. Jeltsch. A high-order boundary treatment for Cartesian-grid methods. *J. Comput. Phys.*, 140:259–277, 1998.
- [22] J. Glimm, X. Li, Y. Liu, Z. Xu, and N. Zhao. Conservative front tracking with improved accuracy. *SIAM J. Numer. Anal.*, 39:179–200, 2003.
- [23] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration : Structure-Preserving Algorithms for Ordinary Differential Equations*, volume 31 of *Springer Series in Comput. Mathematics*. Springer-Verlag, 2nd edition, 2006.
- [24] C. Helzel, M.J. Berger, and R.J. Leveque. A high-resolution rotated grid method for conservation laws with embedded geometries. *SIAM J. Sci. Comput.*, 26:785–809, 2005.
- [25] X. Y. Hu, B. C. Khoo, N. A. Adams, and F. L. Huang. A conservative interface method for compressible flows. *J. Comput. Phys.*, 219(2):553–578, 2006.
- [26] S. R. Idelsohn, J. Marti, A. Limache, and E. Onate. Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM. *Comput. Meth. Appl. Mech. Eng.*, 197(19–20):1762–1776, 2008.
- [27] G-S. Jiang and C-W. Shu. Efficient implementation of Weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.
- [28] Shi Jin and Jian-Guo Liu. The effects of numerical viscosities : I. Slowly moving shocks. *J. Comput. Phys.*, 126:373–389, 1996.
- [29] G. Kalitzin and G. Iaccarino. Turbulence modeling in an immersed-boundary rans method. CTR Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford Univ., 2002.
- [30] G. Kalitzin and G. Iaccarino. Towards immersed boundary simulation of high reynolds number flows. CTR Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford Univ., 2003.
- [31] B. Keen and S. Karni. A second order kinetic scheme for gas dynamics on arbitrary grids. *J. Comput. Phys.*, 205:108–130, 2005.

- [32] Michel Lesoinne and Charbel Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Methods Appl. Mech. Engrg.*, 134:71–90, 1996.
- [33] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31(4):1019–1044, 1994.
- [34] R. J. LeVeque and Z. Li. Immersed interface methods for stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.
- [35] C. Mariotti. Lamb’s problem with the lattice model Mka3D. *Geophys. J. Int.*, 171:857–864, 2007.
- [36] G. H. Miller and P. Colella. A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing. *J. Comput. Phys.*, 183(1):26–82, 2002.
- [37] J. Mohd-Yusof. Combined immersed-boundary/b-spline methods for simulation of flow in complex geometries. CTR Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford Univ., 1997.
- [38] L. Monasse and C. Mariotti. An energy-preserving Discrete Element Method for elastodynamics. submitted.
- [39] W.F. Noh. *Fundamental methods of hydrodynamics*, volume 3 of *Methods of computational physics*, pages 117–179. Academic Press, New York/London, 1964.
- [40] L. Olovsson. *On the arbitrary Lagrangian-Eulerian Finite Element Method*. PhD thesis, Linköping University, 2000.
- [41] A. Paccou, G. Chiavassa, J. Liandrat, and K. Schneider. A penalization method applied to the wave equation. *C. R. Mécanique*, 333(1):79–85, 2005.
- [42] P. De Palma, M. D. de Tullio, G. Pascazio, and M. Napolitano. An immersed-boundary method for compressible viscous flows. *Comp. Fluid*, 35(7):693–702, 2006.
- [43] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield, and M.L. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *J. Comput. Phys.*, 120:278–304, 1995.
- [44] C. S. Peskin. *Flow patterns around heart valves : A digital computer method for solving the equations of motion*. PhD thesis, Albert Einstein College of Medicine, 1972.
- [45] C.S. Peskin. The immersed boundary method. *Acta Numer.*, 11:1–39, 2002.
- [46] S.K. Sambasivan and H.S. UdayKumar. Ghost Fluid method for strong shock interactions Part 2 : Immersed solid boundaries. *AIAA J.*, 47(12):2923–2937, 2009.
- [47] G. Strang. On construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–516, 1968.
- [48] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–173, 1984.
- [49] S. Xu and Z. Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *J. Comput. Phys.*, 216(2):454–493, 2006.